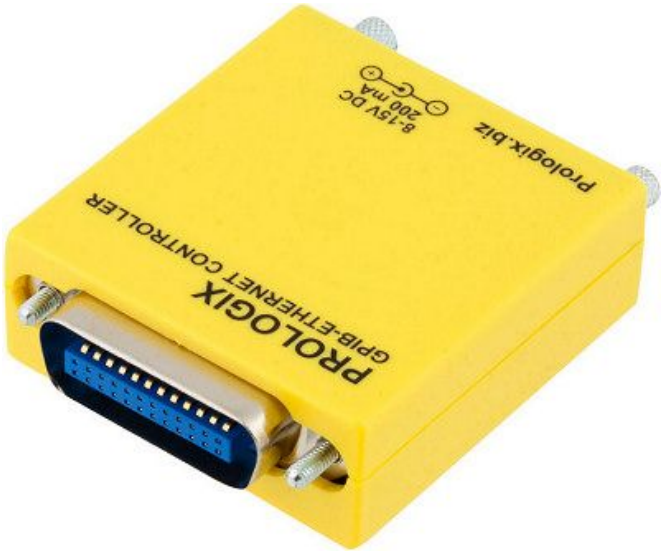
PROLOGIX

GPIB-ETHERNET CONTROLLER



ユーザーマニュアル

バージョン　1.6.6.0

May 14, 2013

PROLOGIX.BIZ

目　次

1. はじめに 4
2. インストール 4
3. ファームウェアのアップグレード 4
4. ホストソフトウェア 4
5. ネットとワークの設定 5
6. GPIBの設定 5
7. 操作モード 5
7.1. コントローラモード 5
7.2. デバイスモード 6
8. データ送信 6
8.1. ２進データ送信 7
9. コマンド 8
9.1. addr 8
9.2. auto 8
9.3. clr 9
9.4. eoi 9
9.5. eos 10
9.6. eot_enable 10
9.7. eot_char 11
9.8. ifc 11
9.9. llo 11
9.10. loc 11
9.11. lon 11
9.12. mode 12
9.13. read 12
9.14. read_tmo_ms 12
9.15. rst 13
9.16. savecfg 13
9.17. spoll 14
9.18. srq 14
9.19. status 14
9.20. trg 15
9.21. ver 15
9.22. help 15
10. 仕　様 15

注記：　この日本語マニュアルは、原文（英文マニュアル）を（株）コンパス・ラブが
和訳したものです。表現に疑義が生じたときは英文マニュアルを参照ください。

変更履歴

2013 年 5 月 14 日	++llo コマンド追記。
2012 年 5 月 7 日	++status 文書のエラー修正
2011 年 4 月 18 日	++lon コマンド追記。
2009 年 9 月 14 日	++savecfgコマンド追記。 read_tmo_msの最大値を3000 msに修正。
2009 年 5 月 22 日	++statusコマンド追記。第２のアドレスを含めるため ++addr, ++spoll, ++trgコマンド記述を更新。 2 進データ送信のセクションを追加。
2008 年 4 月 5 日	最初のバージョン。

1. はじめに

Prologix GPIB-ETHERNET コントローラは、ネットワークポートを持つ全てのコンピュータを GPIB コントローラあるいはデバイスに変えます。

コントローラモードでは、Prologix GPIB-ETHERNET コントローラは、オシロスコープ、ロジックアナライザーとスペクトラムアナライザーのような GPIB 対応機器を遠隔制御できます。

デバイスモードでは、Prologix GPIB-ETHERNET コントローラは、コンピュータを計測器フロントパネルからのデータダウンロードとスクリーン描画のための GPIB 周辺機器に変えます。

両モードで、Prologix GPIB-ETHERNET コントローラは、ホストコンピュータから受信した高レベルコマンドを解釈し、適当な低レベル GPIB プロトコルハンドシェーキングを実行します。

2. インストール

Prologix GPIB-ETHERNET コントローラを、イーサーネットケーブルを用いて、ネットワーク対応コンピュータに接続してください。特別なドライバーは必要としません。使用するケーブルのタイプは、コンピュータに依存します。コンピュータが、ほとんど全ての新しいコンピュータがサポートしている auto-MDIX をサポートしている場合は、ストレートイーサーネットケーブルを使用します。コンピュータが auto-MDIX をサポートしていない場合は、クロスイーサーネットケーブルを使用します。クロスイーサーネットケーブルが使用できない場合あるいは作動しない場合は、Prologix GPIB-ETHERNET コントローラとコンピュータをネットワークハブ（あるいはスイッチ）に接続します。

3. ファームウェアのアップグレード

Prologix GPIB-ETHERNET コントローラファームウェアは、フィールドアップグレードが可能です。最新のファームウェアとしアップグレードインストールが、prologix.biz で利用可能です。

4. ホストソフトウェア

多種多様なホストソフトウェアを、Prologix GPIB-ETHERNET コントローラとの通信に使用できます。

ターミナルプログラム－HyperTerminal、Tera Term Pro あるいは Minicom のようなターミナルエミュレーションプログラムを、コントローラとそれに接続された機器との通信に使用できます。

カスタムアプリケーション－ネットワークアクセスを提供するプログラミング言語あるいは開発環境を、カスタムアプリケーションの開発に使用できます。National Instruments LabView と Agilent VEE のようなグラフィックプログラミング環境を、同様に使用できます。

EZGPIB－データ収集アプリケーション開発のために、Ulrich Bangert により開発された使い易いプログラミング環境です。このツールへのリンクを、prologix.biz で見つけられます。

プロッターエミュレータ－7470.exe のようなプロッターエミュレーションアプリケーションを、Prologix GPIB-ETHERNET コントローラを用いて GPIB 対応機器からスクリーン描画を得るのに使用できます。これらのツールの設定方法の詳細は、prologix.biz で利用可能です。

5. ネットワーク設定

Prologix GPIB-ETHERNET コントローラは、静的 IP アドレスと動的(DHCP) IP アドレスをサポートします。コントローラのネットワークパラメーターを prologix.biz で利用可能なツール **NETFINDER** を用いて設定できます。

6. GPIB の設定

Prologix GPIB-ETHERNET コントローラの GPIB パラメータを、次の方法のいずれかを用いて設定できます:

Prologix.exe－Prologix.exe は、コントローラ設定のために John Miles により開発されたオープンソースツールです。ツールへのウェブリンクは、prologix.biz で見出せます。

ターミナルプログラム－HyperTerminal、Tera Term Pro あるいは Minicom のようなターミナルエミュレーションプログラムを、適当なコマンドの手動入力によりコントローラの設定に使用できます。ターミナルプログラムの使用は、Prologix GPIB-ETHERNET コントローラの IP アドレスの TCP 接続を、ポート 1234 としてオープンします。入力コマンドを見るためにターミナルプログラムの“Local Echo”を可能にすることが出来ます。種々のターミナルプログラムの詳細設定ステップには、prologix.biz の FAQ、あるいはプログラムユーザーマニュアルを参照下さい。

Prologix GPIB-ETHERNET コントローラは、不揮発性メモリーに最新構成設定を保存します。これらの設定のアドレス仕様はありません。異なる構成設定を必要とする GPIB バスの複数の機器を持っている場合は、各機器と通信する前に設定を変更しなければなりません。

7. 操作モード

Prologix GPIB-ETHERNETコントローラを、CONTROLLERとDEVICEの2つのモードで操作できます。++mode コマンド(コマンド参照)を用いて、2 つのモード間を切り替えられます。

7.1 コントローラモード

コントローラモードでは、GPIB-ETHERNET コントローラは、GPIB バスの Controller-In-Charge (CIC)として作動します。コントローラがネットワークターミネーター CR (ASCII 13)あるいは LF (ASCII 10) でターミネートされたネットワークポート上のコマンドを受信すると、リスンのために、現在指定されているアドレス(++addr コマンド参照)の GPIB インスツルメントをアドレス指定し、受信データを伝えます。

Read-After-Write機能が利用可能になっていると(++autoコマンド参照)、Prologix GPIB-ETHERNETコントローラは、その応答を読むために、コマンド送信後、トークのためにそのインスツルメントをアドレス指定します。GPIB上のインスツルメントから受信した全てのデータを、ネットワークでホストに送信します。Read-After-Write機能は、インスツルメントとの交信を簡単にします。低レベルのGPIBプロトコル詳細を考慮しないで、コマンドを送信し応答を読めます。

Read-After-Write機能が利用可能でない時は、Prologix GPIB-ETHERNETコントローラは、トークのためのインスツルメントを自動的にアドレス指定しません。データを読むには、++read コマンドを使用しなければなりません。

コントローラモードは、インスツルメントの遠隔制御とホストコンピュータからのプロットコマンドの送信によって、スクリーンプロットをダウンロードするのに使用します。

7.2 デバイスモード

デバイスモードでは、Prologix GPIB-ETHERNETコントローラは、GPIBバスの別の周辺機器として作動します。このモードでは、コントローラは、GPIBトーカあるいはGPIBリスナとしてのみ作動可能です。Prologix GPIB-ETHERNETコントローラは、このモードの間、Controller-In-Chargeではないので、GPIBコントローラからのコマンドの受信を想定しています。デバイスモードがenableの場合、Prologix GPIB-ETHERNETコントローラは、GPIBリスナとして自身を設定します。GPIBポート上の、コントローラで受信した全てのデータは、バッファリングなしにネットワークポートに伝えられます。

ネットワーク上のホストから受信した全てのデータを、GPIBコントローラが、トークするためにPrologix GPIB-ETHERNETコントローラをアドレス指定するまでバッファし、その時には、バッファしたデータをGPIBポートに伝えます。コントローラは、1つのコマンドだけをバッファリングできます。以前のものがまだGPIB上に送られていない場合は、ネットワーク上で受信した次のコマンドは、以前にバッファリングしたものを上書きします。

デバイスモードは、プロッターエミュレーションソフトウェアを用いて表現するために、インスツルメントのフロントパネルから、スクリーンプロットをダウンロードするのに使用します。

8. データ通信

コントローラとデバイスモードでは、ETHERNETポート上で受信した文字は内部バッファに集められ、ETHERNET終了文字CR (ASCII 13) あるいはLF (ASCII 10)を受信した時、解釈します。もしCR、LF、ESC (ASCII 27)あるいは ‘+’ (ASCII 43) 文字がETHERNETデータの一部であるなら、それにESC文字を先行して、回避しなければなりません。全ての回避されなかったETHERNETデータのLF、CR とESCと ‘+’ 文字は、廃棄されます。

前述したように、回避されなかったCRやLFは、ETHERNETターミネータとして作動します。CR やLFは除かれ、GPIB終了文字(++eosコマンドで指定される)が、インスツルメントへのデータ送信前に追加されます。

回避されなかった“++”文字で始まる全てのETHERNET入力は、コントローラコマンドとして解釈し、GPIB上に送信されません。

そのように（++autoコマンドか++read コマンドを使用して、）設定した時は、インスツルメントから受信した文字をホストに送信します。インスツルメントにデータを送信する時と違って、文字の置換はありません。

++eot_char コマンドを、GPIB EOI 信号確認の検出に使用できます。

8.1 バイナリーデータ通信

Prologix GPIB-ETHERNET コントローラは、GPIB で使用可能なインスツルメントとのバイナリーデータの送受信が可能です。

インスツルメントからのバイナリーデータの受信には、特別な処置は必要ありません。インスツルメントから受信した全てのバイナリーデータは、PC にイーサネットで、ASCII データと同様に未修正で送信されます。インスツルメントからのバイナリーデータは、通常、CR や LF 文字で終了しないので(通常、ASCII データの場合のように)、データの終わりを示す EOI を検出する++eot_enable コマンドの使用を希望することができます。より詳細は、++eot_enable コマンドのヘルプをご覧下さい。

インスツルメントへのバイナリーデータ送信時は、特別の注意をしなければなりません。次の文字の何れかがバイナリーデータに発生した場合－CR (ASCII 13)、LF (ASCII 10)、ESC (ASCII 27)、‘+’ (ASCII 43) －それらに ESC 文字を先行することによって、それらを避けなければなりません。

例えば、次の(10 進法)バイナリーデータの送信に対して:

00 01 02 13 03 10 04 27 05 43 06

それを次のように避けなければなりません:

00 01 02 **27** 13 03 **27** 10 04 **27** 27 05 **27** 43 06

更に、CR や LF のような GPIB 終了文字をバイナリーデータに追加した場合、大部分のインスツルメントは混同します。そのような挙動を抑制するために、++eos 3 コマンドを使用します。より詳細は、++eos コマンドのヘルプをご覧下さい。

9. コマンド

Prologix GPIB-ETHERNETコントローラは、その挙動を設定するいくつかのコマンドを提供します。それらを、次の節で詳細に説明します。全てのコマンドは“++”文字列で始まります。

9.1 addr

addrコマンドは、GPIBアドレスの設定、あるいは問い合わせに使用します。GPIBアドレスの意味は、コントローラの動作モードに依存します。コントローラモードでは、制御しているインスツルメントのGPIBアドレスを指します。デバイスモードでは、Prologix GPIB-ETHERNETコントローラがエミュレートしているGPIB周辺機器のアドレスです。

オプションのセカンダリーアドレスも指定できます。セカンダリーアドレスは、プライマリーアドレスとスペース文字で分離しなければなりません。有効なセカンダリーアドレス値は96から126(10進)です。96のセカンダリーアドレスは、0のセカンダリーGPIBアドレスに対応し、97は、1に対応するなど。セカンダリーアドレスの指定は、デバイスモードに影響しません。

このコマンドをパラメータなしで出すと、現在設定されているアドレス(指定されている場合、プライマリーとセカンダリー)を返します。

構文：++addr [<PAD> [<SAD>]]
PAD(プライマリーアドレス)は、0 と 30 の間の 10 進値です。
SAD(セカンダリーアドレス)は、96 と 126 の間の 10 進値です。SAD はオプションです。

利用可能モード:コントローラ、デバイス

例:

```
++addr 5      ープライマリーアドレスを 5 にセットします。
++addr        ー現在のアドレスを問合せます。
++addr 9 96   ープライマリーアドレスを 9 に、セカンダリーアドレスを 0 にセットします。
```

注意:
多くの HP-GL/2 プロッターの初期値 GPIB アドレスは 5 です。

9.2 auto

Prologix GPIB-ETHERNETコントローラは、それらの応答を読むコマンドを送信後、トークのために、インスツルメントを自動的にアドレスするよう設定できます。Read-After-Writeと呼ばれる機能は、繰り返し読込コマンドを出さねばならぬことからユーザーを救います。このコマンドは、Read-After-Write 機能を使用可能、あるいは使用不可にします。

更に、autoコマンドは、トークあるいはリスンのために、現在指定されているアドレスのインスツルメントをアドレス指定します。++auto 0 は、リスンのためにインスツルメントをアドレスし、++auto 1はトークのためにインスツルメントをアドレスします。

もし、このコマンドを引数なしで出した場合、read after-write 機能の現在の状態を返します。

構文：++auto [0|1]

利用可能モード:コントローラ

注記:
いくつかのインスツルメントは、応答を生成しないコマンド(しばしばnon-queryコマンドと呼ばれる)を送信後、トークのためにアドレス指定すると、"Query Unterminated" あるいは "420" エラーを生じます。事実上インスツルメントは、私は尋ねられていますが話すことはありませんと、言っています。このエラーはしばしば良性で無視できます。さもなければ、インスツルメントの応答を読むために、++readコマンドを使用して下さい。

例えば：
++auto 0　　－ read-after-writeをオフにし、インスツルメントをリスンのためにアドレスします。
SET VOLT 1.0　－ non-query コマンド
*idn?　　－ query コマンド
++read eoi　－ インスツルメントにより EOI を示されるまで読む
"HP54201A"　－ インスツルメントからの応答

9.3 clr

このコマンドは、Selected Device Clear (SDC) メッセージを現在指定されているGPIBアドレスに送信します。特別なインスツルメントがこのメッセージにどのように応答するかについての詳細は、プログラミングマニュアルを参照して下さい。

構文：++clr

利用可能モード:コントローラ

9.4 eoi

このコマンドは、GPIBポート上に送る全てのコマンドの最後の文字と共に、EOI信号の宣言を使用可か使用不可にします。いくつかのインスツルメントは、コマンドの終了を適切に検出するために、EOI信号を宣言する必要があります。

構文：++eoi [0|1]

利用可能モード:コントローラ、デバイス

例:

++eoi 1	最後の文字と共に、EOI 宣言を使用可にします
++eoi 0	EOI 宣言を使用不可にします
++eoi	EOI 宣言を使用可、使用不可を問い合わせます

9.5 eos

このコマンドは、GPIBターミネート文字を指定します。ホストからのデータをネットワーク上で受信すると、全てのnon-escaped LF、CR、ESC文字を除去し、インスツルメントにデータを送信する前に、このコマンドで指定されたGPIBターミネータを追加します。このコマンドは、GPIBポートで受信したインスツルメントからのデータに影響しません。

このコマンドが引数なしに出された場合、現在の設定を返します。

構文：++eos [0|1|2|3]　ここで: 0 – CR+LF, 1 – CR, 2 – LF, 3 – なし

利用可能モード:コントローラ、デバイス

例:

++eos 0	インスツルメントコマンドに CR+LF を追加します
++eos 1	インスツルメントコマンドに CR を追加します
++eos 2	インスツルメントコマンドに LF を追加します
++eos 3	インスツルメントコマンドに何も追加しません
++eos	現在の EOS の状態を問合せます

9.6 eot_enable

このコマンドは、GPIBポートから文字を読み込み中に、EOIが検出された時はいつでも、ネットワーク出力へユーザーが指定した文字(eot_char参照) の追加を利用可あるいは利用不可にします。

このコマンドを、引数なしに出した場合、eot_enable の現在の状態を返します。

構文：eot_enable [0|1]

利用可能モード:コントローラ、デバイス

例:

++eot_enable 1	EOI 検出時、ユーザー指定文字を追加します
++eot_enable 0	EOI 検出時、文字を追加しません
++eot_enable	現在の eot_enable の状態を問合せます

9.7 eot_char

このコマンドは、eot_enableが1にセットされかつEOIを検出した時に、ネットワーク出力に追加される文字を指定します。

もし、引数なしでこのコマンドを出した場合、現在指定されている文字を返します。

構文：eot_char [<char>]　ここで、<char> は 256 未満の 10 進値

利用可能モード:コントローラ、デバイス

例:

++eot_char 42	EOI を検出した時、* (ASCII 42)を追加します。
++eot_char	現在設定されている eot_char を問い合わせます。

9.8 ifc

このコマンドは、Prologix GPIB-ETHERNE コントローラを GPIB バス Controller-In-Charge にして、150 ミリ秒、GPIB IFC 信号を宣言します。

構文：++ifc

利用可能モード:コントローラ

9.9. llo

このコマンドは、現在アドレス指定されているインスツルメントのフロントパネル操作を不可能にします。

構文:++llo

利用可能モード：コントローラ

9.10. loc

このコマンドは、現在アドレス指定されているインスツルメントのフロントパネル操作を可能にします

構文：++loc

利用可能モード:コントローラ

9.11. lon

このコマンドは、現在指定されているアドレスに関わりなく、GPIB バスの全てのトラフィックをリスンすべく GPIB-ETHERNET を設定します。この設定は、“リスンオンリー” モード として知られています。このモードでは、コントローラは受信できるだけで、データ送信で きません。

構文：lon [0|1]

利用可能モード:デバイス

例:

++lon 1	“リスンオンリー”モードにします
++lon 0	“リスンオンリー”モードを使用不可にします
++lon	“リスンオンリー”モードを問合せます

9.12. mode

このコマンドは、Prologix GPIB-ETHERNETコントローラをコントローラまたはデバイスに設定します。

もし、引数なしにこのコマンドを出すと、現在のモードを返します。

構文：++mode [0|1]　ここで: 1-コントローラ, 0-デバイス

利用可能モード:コントローラ、デバイス

例:

++mode 1	コントローラモードに切り替えます
++mode 0	デバイスモードに切り替えます
++mode	現在のモードを問合せます

9.13 read

このコマンドは、以下の時までインスツルメントからのデータを読むのに使用できます:

- EOI を検出するか、タイムアウト終了になる、または
- 指定された文字を読み込むか、タイムアウト終了になる、または
- タイムアウト終了になる

タイムアウトはread_tmo_msコマンドでセットし、inter-character delayすなわち、最後の文字を読み込んでからの遅れに適用します。タイムアウトは、データ読込みの合計時間と一緒になりません。

構文：++read [eoi|<char>]　ここで: <char> は 256 未満の 10 進値です。

利用可能モード:コントローラ

例:

++read	タイムアウトまで読みます
++read eoi	EOI を検出するか、タイムアウトまで読みます
++read 10	LF (ASCII 10)を受信するかタイムアウトまで読みます

9.14. read_tmo_ms

このコマンドは、read コマンドと spoll コマンドで使用するため、ミリ秒で、タイムアウト値を指定します。タイムアウトを、1 と 3000 ミリ秒間の全ての値にセットできます。

構文：++read_tmo_ms <time>　ここで<time> は 1 と 3000 の間の 10 進値です

利用可能モード:コントローラ

9.15. rst

このコマンドは、コントローラのパワーオンリセットを行います。処理に約 5 秒かかります。この時間中にネットワークで受信した全ての入力は無視されます。

構文：++rst

利用可能モード:コントローラ、デバイス

9.16. savecfg

このコマンドは、EPROMの設定パラメータの自動保存を使用可にするか、使用不可にします。
使用可の場合、次の設定パラメータは、アップデートされるといつでも保存されますー
mode, addr, auto, eoi, eos, eot_enable, eot_char, read_tmo_ms。

しかし、頻繁なアップデートは、結局はEPROMを消耗します。このコマンドは、EPROMの消耗を減らすために、設定パラメータの自動保存を一時的に使用不可にするために使用します。

Savecfg 設定自体はEPROMに保存されません。それはスタート時に常に使用可です（電源オンまたはリセット後）。

構文：++savecfg [0|1]

利用可能モード:コントローラ、デバイス

例:

++savecfg 1	EPROM の設定パラメータの保存を使用可にします
++savecfg 0	EPROM の設定パラメータの保存を使用不可にします
++savecfg	現在の設定を問合せます

注記：
“++savecfg 1”コマンドは、パラメータの自動保存を使用可能にするだけでなく、全ての設定パラメータの現在値をすぐに保存します。

9.17. spoll

このコマンドは、指定されたアドレスのインスツルメントのシリアルポールを実行します。アドレスが指定されない場合、このコマンドは、現在アドレス指定されているインスツルメント（前の++addrコマンドでセットした）をシリアルポールします。このコマンドは、read_tmo_msコマンドで指定されたタイムアウトを使用します。

構文: ++spoll [<PAD> [<SAD>]]
PAD（プライマリーアドレス）は、0と30の間の10進値です。
SAD（セカンドリーアドレス）は、96と126の間の10進値です。SADはオプションです。

利用可能モード:コントローラ

例:

++spoll 5	プライマリーアドレス5のインスツルメントをシリアルポールします。
++spoll 9 96	プライマリーアドレス9、セカンドリーアドレス0のインスツルメントをシリアルポールします。
++spoll	現在指定されているインスツルメントをシリアルポールします。

9.18. srq

このコマンドは、GPIB SRQ信号の現在の状態を返します。このコマンドは、SRQ 信号が宣言されていれば（Low） ‘1’ を、信号が宣言されていなければ（High） ‘0’ を返します。

構文：++srq

利用可能モード:コントローラ

9.19. status

statusコマンドは、GPIBコントローラによってシリアルポールされた時、返すべきデバイスステータスバイトを指定するのに使います。ステータスバイトのRQSビット(ビット#6)がセットされると、SRQ信号が宣言(low)状態になります。シリアルポール後、SRQラインは宣言無し状態になり、ステータスバイトを0にセットされます。ステータスバイトは、電源オン時に0に初期化されます。

GPIBコントローラからDEVICE CLEAR (DCL)メッセージ、あるいはSELECTED DEVICE CLEAR (SDC) メッセージを受けるとSRQも宣言状態でなくなり、ステータスバイトはクリアされます。

もし、このコマンドが何の引数もなく出されると、現在指定されているステータスバイトを返します。

構文：++status [0-255]

利用可能モード：デバイス

例:

++status 72　72にシリアルポールステータスバイトを指定します。ビット#6がセットされ、このコマンドはSRQを宣言します。

++status　現在のシリアルポールステータスバイトを問い合わせます。

9.20. trg

このコマンドは、指定されたアドレスのデバイスにGroup Execute Trigger GPIBコマンドを出します。15アドレスまで指定できます。アドレスはスペースで区切らなければなりません。もし、アドレスを指定しない場合、Group Execute Triggerコマンドを、現在アドレス指定されたインスツルメント(前の++addr コマンドでセットした)に出します。
Group Execute Triggerコマンドへの指定したインスツルメントの応答に対しては、プログラミングマニュアルを参照して下さい。

構文：++trg [<PAD1> [<SAD1>] <PAD2> [SAD2] … <PAD15> [<SAD15>]]

利用可能モード:コントローラ

9.21. ver

このコマンドは、Prologix GPIB-ETHERNET コントローラのバージョン文字列を返します。
構文：++ver

利用可能モード:コントローラ、デバイス

9.22. help

このコマンドは、利用可能な全コマンドの概要を印字(画面)します。
構文：++help

利用可能モード:コントローラ、デバイス

10. 仕　様

サポート OS　：Windows 98/2000/XP/Vista/7、Mac OS 8/9/X、Linux、FreeBSD
サポート規格　：IEEE 488.1、IEEE 488.2
サポートされない GPIB コマンド　：PARALLEL POLL、 PASS CONTROL
電源　：8-15V DC、200 mA
インジケーター　：電源
IP 設定　：静的と動的(DHCP)
TCP ポート：1234
寸法　：2.5 in. (L) x 2.5 in. (W) x 1.0 in. (H)
重量　：3 oz。